AliCoCo: Alibaba E-commerce Cognitive Concept Net

Xusheng Luo*
Alibaba Group, Hangzhou, China
lxs140564@alibaba-inc.com

Luxin Liu, Le Bo, Yuanpeng Cao,
Jinhang Wu, Qiang Li
Alibaba Group, Hangzhou, China

Yonghua Yang, Keping Yang
Alibaba Group, Hangzhou, China

Kenny Q. Zhu
Shanghai Jiao Tong University, Shanghai, China

ABSTRACT

One of the ultimate goals of e-commerce platforms is to satisfy various shopping needs for their customers. Much efforts are devoted to creating taxonomies or ontologies in e-commerce towards this goal. However, user needs in e-commerce are still not well defined, and none of the existing ontologies has the enough depth and breadth for universal user needs understanding. The semantic gap in-between prevents shopping experience from being more intelligent. In this paper, we propose to construct a large-scale e-commerce **Co**gnitive **Co**ncept net named "**AliCoCo**", which is practiced in **Ali**baba, the largest Chinese e-commerce platform in the world. We formally define user needs in e-commerce, then conceptualize them as nodes in the net. We present details on how AliCoCo is constructed semi-automatically and its successful, ongoing and potential applications in e-commerce.

KEYWORDS

Concept Net; E-commerce; User Needs

ACM Reference Format:
Xusheng Luo, Luxin Liu, Le Bo, Yuanpeng Cao, Jinhang Wu, Qiang Li, Yonghua Yang, Keping Yang, and Kenny Q. Zhu. 2020. *AliCoCo*: Alibaba E-commerce Cognitive Concept Net . In *2020 International Conference on Management of Data (SIGMOD '20), June 14–19, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3357384.3357812

*Corresponding author

SIGMOD '20, June 14–19, 2020, Portland, OR, USA

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357812

1 INTRODUCTION

One major functionality of e-commerce platforms is to match the shopping need of a customer to a small set of items from an enormous candidate set. With the rapid developments of search engine and recommender system, customers are able to quickly find those items they need. However, the experience is still far from "intelligent". One significant reason is that there exists a huge semantic gap between what users need in their mind and how the items are organized in e-commerce platforms. The taxonomy to organize items in Alibaba (actually almost every e-commerce platforms) is generally based on **CPV** (Category-Property-Value): thousands of categories form a hierarchical structure according to different granularity, and properties such as color and size are defined upon each leaf node. It is a natural way of organizing and managing billions of items in nowadays e-commerce platform, and already becomes the essential component in downstream applications including search and recommendation. However, existing taxonomies or ontologies in e-commerce are difficult to interpret various user needs comprehensively and accurately due to the semantic gap, which will be explained in the following two scenarios.

For years, e-commerce search engine is teaching our users how to input keywords wisely so that the wanted items can be quickly found. However, it seems keyword based searching only works for those users who know the exact product they want to buy. The problem is, users do not always know the exact product. More likely what they have in mind is a type or a category of products, with some extra features. Even worse, they only have a scenario or a problem but no idea what items could help. In these cases, a customer may choose to conduct some research outside the e-commerce platform to narrow down to an exact product, which harms the user experience and making e-commerce search engine not intelligent at all. If tracing back to the source, the real reason behind this is that existing ontologies in e-commerce doesn't contain structured knowledge indicating what products are needed for an "outdoor barbecue" or what is "preventing the olds from getting lost". Typing search queries like these inevitable leads to user needs mismatch and query understanding simply degenerates to key words matching.

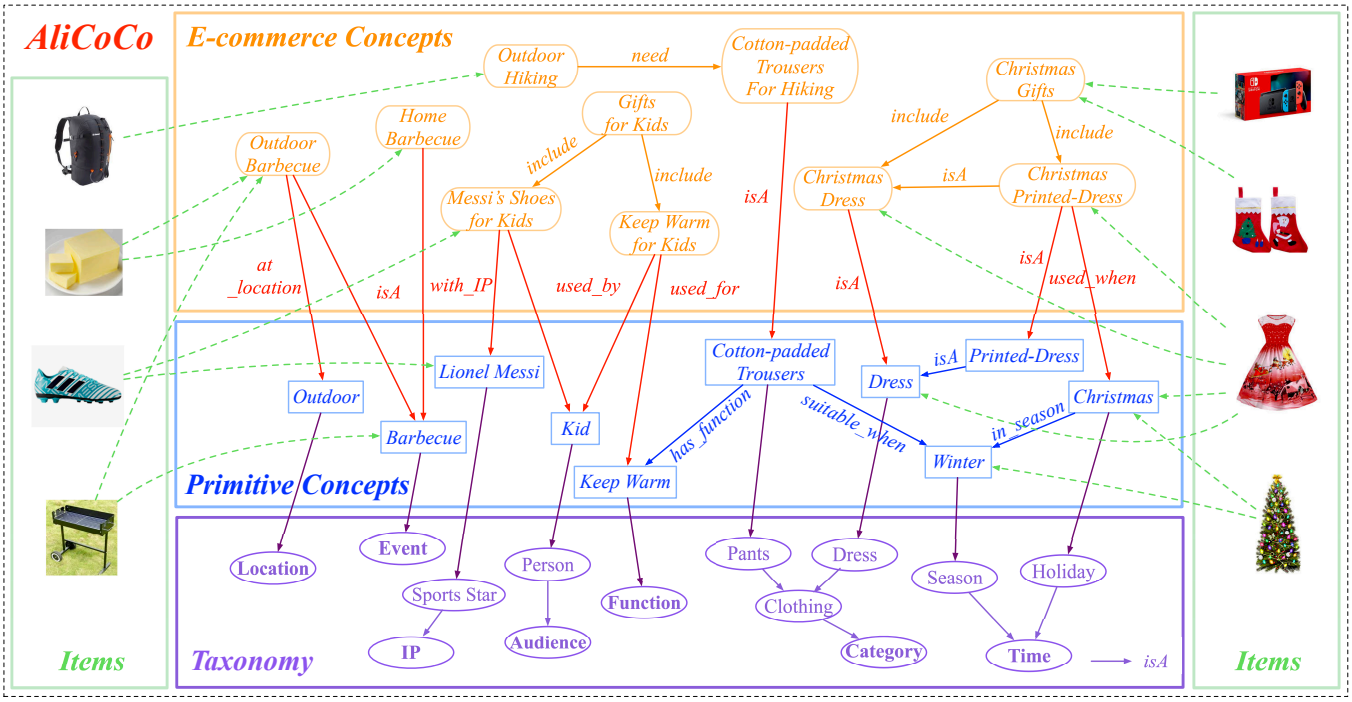

Figure 1: Overview of "AliCoCo", which consists of four layers: e-commerce concepts, primitive concepts, taxonomy and items.

The same problem exists in item recommendation. Due to the prohibitive size of transaction data in real-world industry scenario, recommendation algorithms widely adopt the idea of item-based CF [24], which can recommend from very large set of options with relatively small amount of computation, depending on the pre-calculated similarity between item pairs. The recommender system uses user's historical behaviors as triggers to recall a small set of most similar items as candidates, then recommends items with highest weights after scoring with a ranking model. A critical shortcoming of this framework is that it is not driven by user needs in the first place, which inevitably leads to a dilemma where items recommended are hard to be explained except for trivial reasons such as "similar to those items you have already viewed or purchased". Besides, it also prevents the recommender system from jumping out of historical behaviors to explore other implicit or latent user interests. Therefore, despite the widespread of its use, the performance of current recommendation systems is still under criticism. Users are complaining that some recommendation results are redundant and lack novelty, since current recommender systems can only satisfy very limited user needs such as the needs for a particular category or brand. The lack of intermediate nodes in current e-commerce ontologies that can represent various user needs constrains the development of recommender systems.

In this paper, we attempt to bridge the semantic gap between actual user needs and existing ontologies in e-commerce platforms by building a new ontology towards universal user needs understanding. It is believed that the cognitive system of human beings is based on *concepts* [4, 20], and the taxonomy and ontology of concepts give humans the ability to understand [30]. Inspired by it, we construct the ontology mainly based on concepts and name it "**AliCoCo**": **Co**gnitive **Co**ncept Net in **Ali**baba. Different from most existing e-commerce ontologies, which only contain nodes such as categories or brands, a new type of node, e.g., "Outdoor Barbecue" and "Keep Warm for Kids", is introduced as bridging concepts connecting user and items to satisfy some high-level user needs or shopping scenarios. Shown in the top of Figure 1, we call these nodes "**e-commerce concepts**", whose structure represents a set of items from different categories with certain constraints (more details in Section 5) . For example, "Outdoor Barbecue" is one such e-commerce concept, consisting of products such as grills, butter and so on, which are necessary items to host a successful outdoor barbecue party. Therefore, AliCoCo is able to help search engine directly suggest a customer "items you will need for outdoor barbecue" after he inputs keyword "barbecue outdoor", or help recommender system remind him of preparing things that can "keep warm for your kids" as there will be a snowstorm coming next week.

There are several possible practical scenarios in which applying such e-commerce concepts can be useful. The first and most natural scenario is directly displaying those concepts to users together with its associated items. Figure 2(a/b) shows the real implementation of this idea in *Taobao* [1] App. Once a user typing "Baking" (a), he will enter into a page (right) where different items for baking are displayed, making the search experience a bit more intelligent. It can also be integrated into recommender systems. Among normal recommended items, concept "Tools for Baking" is displayed to users as a card with its name and the picture of a representative item (b). Once a user clicks on it, he will enter into the page on the right. In this way, the recommender system is acting like a salesperson in a shopping mall, who tries to guess the needs of his customer and and then suggests how to satisfy them. If their needs are correctly inferred, users are more likely to accept the recommended items. Other scenarios can be providing explanations in search or recommendation as shown in Figure 2(c). While explainable recommendation attracts much research attention recently [33], most existing works are not practical enough for industry systems, since they are either too complicated (based on NLG [8, 32]), or too trivial (e.g., "how many people also viewed" [9, 17]). Our proposed concepts, on the contrary, precisely conceptualize user needs and are easy to understand.

- We claim that current ontologies in e-commerce platforms are unable to represent and understand actual user needs well and therefore prevent shopping experience from being more intelligent. To bridge the semantic gap in between, we formally define user needs in e-commerce and propose to build an end-to-end large comprehensive knowledge graph called "AliCoCo", where the "concept" nodes can explicitly represent various shopping needs for users.
- To construct such a large-scale knowledge graph, we adopt a semi-automatic way by combining both machine learning efforts and manual efforts together. We detailed introduce the four-layer structure of AliCoCo and five non-trivial technical components. For each component, we formulate the problem, point out the challenge, describe effective solutions and give thorough evaluations.
- AliCoCo is already gone into production in Alibaba, the largest e-commerce platform in China. It benefits a series of applications including search and recommendation. We believe the idea of user needs understanding can be further applied in more e-commerce productions. There is ample room for imagination and further innovation in "user-needs driven" e-commerce.

[1] http://www.taobao.com

Figure 2: Three real examples of user-needs driven recommendation. (a): Queries trigger concept cards in semantic search. (b): Display concepts directly to users as cards with a set of related items. (c): Concepts act as explanations in search and recommendation.

The rest of paper is organized as follows: First we give an overview of AliCoCo (Section 2), then present how we construct each of the four layers: Taxonomy (Section 3), Primitive Concepts (Section 4), E-commerce Concepts (Section 5), and Item Associations (Section 6). Section 7 shows overall statistics of AliCoCo and evaluations of five main technical modules. Then we discuss some successful, ongoing and potential applications in Section 8. Section 9 mentions related works, and finally, Section 10 gives a conclusion and delineates possible future work.

2 OVERVIEW

AliCoCo provides an alternative to describing and understanding user needs and items in e-commerce within the same, universal framework. As shown in Figure 1, AliCoCo consists of four components: **E-commerce Concepts**, **Primitive Concepts**, **Taxonomy** and **Items**.

As the core innovation, we represent various user needs as **E-commerce Concepts** (orange boxes) in the top layer of Figure 1. E-commerce concepts are short, coherent and plausible phrases such as "outdoor barbecue", "Christmas

gifts for grandpa" or "keep warm for kids", which describe specific shopping scenarios. User needs in e-commerce are not formally defined previously, hierarchical categories and browse nodes [2] are usually used to represent user needs or interests [34]. However, we believe user needs are far broader than categories or browse nodes. Imaging a user who is planning an outdoor barbecue, or who is concerned with how to get rid of a raccoon in his garden. They have a situation or problem but do not know what products can help. Therefore, user needs are represented by various concepts in AliCoCo, and more details will be introduced in Section 5.

To further understand high-level user needs (aka. e-commerce concepts), we need a fundamental language to describe each concept. For example, "outdoor barbecue" can be expressed as "*<Event: Barbecue>* | *<Location: Outdoor>* | *<Weather: Sunny>* | ...". Therefore, we build a layer of **Primitive Concepts**, where "primitive" means concept phrases in this layer are relatively short and simple such as "barbecue", "outdoor" and "sunny" (blue boxes in Figure 1), comparing to e-commerce concepts above which are compound phrases in most cases. To categorize all primitive concepts into classes, a **Taxonomy** in e-commerce is also defined, where classes with different granularities form a hierarchy via *isA* relations. For instance, there is a path top-down being "*Category->ClothingAndAccessory->Clothing->Dress*" in the taxonomy (purple ovals in Figure 1) .

We also define a schema on the taxonomy, to describe relations among different primitive concepts. For example, there is a relation "*suitable_when*" defined between "class: *Category-Clothing->Pants*" and "class: *Time->Season*", so the primitive concept "cotton-padded trousers" is "suitable_when" the season is "winter".

In the layer of **Items**, billions of items [3] on Alibaba are related with both primitive concepts and e-commerce concepts. Primitive concepts are more like the properties of items, such as the color or the size. However, the relatedness between e-commerce concepts and items represents that certain items are necessary or suggested under a particular shopping scenario. As the example shown in Figure 1, items such as grills and butter are related to the e-commerce concept "outdoor barbecue", while they can not be associated with the primitive concept "outdoor" alone.

Overall, we represent user needs as e-commerce concepts, then adopt primitive concepts with a class taxonomy to describe and understand both user needs and items in the same framework. Besides, e-commerce concepts are also associated directly with items, to form the complete structure of AliCoCo.

[2]https://www.browsenodes.com/

[3]Items are the smallest selling units on Alibaba. Two iPhone Xs Max (each of them is an item) in two shops have different IDs.

3 TAXONOMY

The taxonomy of AliCoCo is a hierarchy of pre-defined classes to index million of (primitive) concepts. A snapshot of the taxonomy is shown in Figure 3. Great efforts from several domain experts are devoted to manually define the whole taxonomy. There are 20 classes defined in the first hierarchy, among which there are 7 classes are specially designed for e-commerce, including "*Category*", "*Brand*", "*Color*", "*Design*", "*Function*", "*Material*", "*Pattern*", "*Shape*" "*Smell*", "*Taste*" and "*Style*", where the largest one is "*Category*" having nearly 800 leaf classes, since the categorization of items is the backbone of almost every e-commerce platform. Other classes such as "*Time*" and "*Location*" are more close to general-purpose domain. One special class worth mentioning is "*IP*" (Intellectual Property), which contains millions of real world entities such as famous persons, movies and songs. Entities are also considered as primitive concepts in AliCoCo. The 20 classes defined in the first hierarchy of the taxonomy are also called "domains".

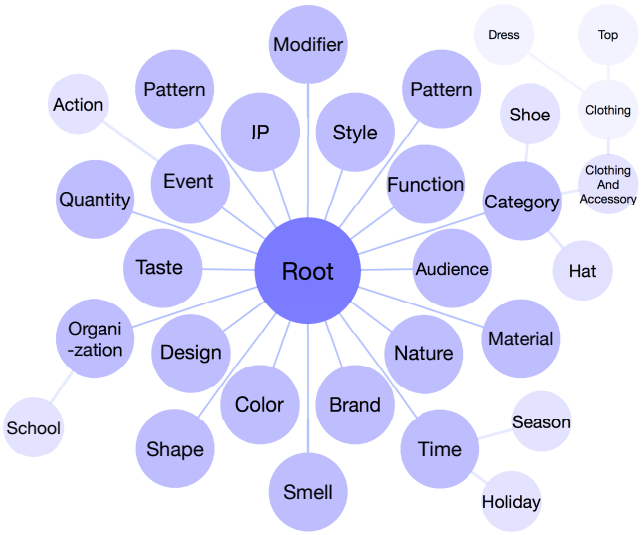

Figure 3: Overview of the taxonomy in AliCoCo.

4 PRIMITIVE CONCEPTS

Primitive concepts with a class taxonomy are expected to describe every item and user need in e-commerce accurately and comprehensively. They are the fundamental building blocks for understanding high-level shopping needs of our customers. In this section, we mainly introduce how we mine these raw primitive concepts (can be seen as vocabulary) and then organize them into the hierarchical structure.

4.1 Vocabulary Mining

There are two ways of enlarging the size of primitive concepts once the taxonomy is defined. The first one is to incorporate existing knowledge from multiple sources through ontology matching. In practice, we mainly adopt rule-based matching algorithms, together with human efforts to manually align the taxonomy of each data source. Details will not be introduced in this paper.

The second one is to mine new concepts from large-scale text corpus generated in the domain of e-commerce such as search queries, product titles, user-written reviews and shopping guides. Mining new concepts of specific classes can be formulated as *sequence labeling* task, where the input is a sequence of words and the output is a sequence of predefined labels. However, the hierarchical structure of our taxonomy is too complicated for this task, so we only use the 20 first-level classes as labels in practice.

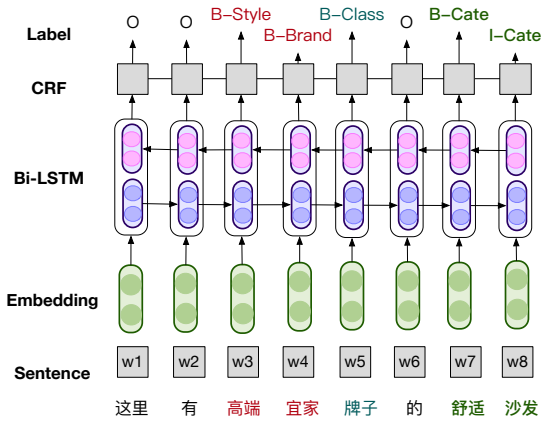

Figure 4: Principle architecture of a BiLSTM-CRF model

Figure 4 shows the principle architecture of a BiLSTM-CRF model, which is the state-of-the-art model for various sequence labeling tasks [14, 23]. BiLSTM-CRF model consists of a BiLSTM layer and a CRF layer, where BiLSTM (Bidirectional-LSTM) enables the hidden states to capture both historical and future context information of the words and CRF (Conditional Random Field) considers the correlations between the current label and neighboring labels.

All the automatically mined *concept: class* pairs are then manually checked to ensure the correctness. Details will be introduced in Section 7.2. Once the class is determined, a surface form then becomes a true primitive concept, and each concept will be assigned a unique ID. There can be several primitive concepts with the same name but different IDs (meanings), giving AliCoCo the ability to disambiguate raw texts.

4.2 Hypernym Discovery

Once primitive concepts of 20 first-level classes (domains) are mined, we continue to classify each primitive concept into fine-grained classes within each domain. In each domain, this task can be formulated as *hypernym discovery*, where we have to predict the hyponym-hypernym relations between arbitrary pair of primitive concepts. In practice, we exploit a combination of two methods: an unsupervised pattern-based method and a supervised projection learning model.

4.2.1 Pattern based. The pattern-based method for hypernym discovery was pioneered by Hearst [12], who defined specific textual patterns like "*Y such as X*" to mine hyponym-hypernym pairs from corpora. This approach is known to suffer from low recall because it assumes that hyponym-hypernym pairs co-occur in one of these patterns, which is often not true when matching the patterns in corpora. Besides those patterns, we adopt other rules to directly discover hypernyms using some special grammar characteristics of Chinese language such as "XX裤 (XX pants)" must be a "裤 (pants)", etc.

4.2.2 Projection learning. The general idea of projection learning is to learn a function that takes as input the word embedding of a possible hyponym p and a candidate hypernym h and outputs the likelihood that there is a hypernymy relationship between p and h. To discover hypernyms for a given hyponym p, we apply this decision function to all candidate hypernyms, and select the most likely ones. Given a pair of candidate p and h, we first obtain their word embeddings $\boldsymbol{p}$ and $\boldsymbol{h}$ through a lookup table where embeddings are pertained on e-commerce corpus. Then we use a projection tensor $\boldsymbol{T}$ to measure how possible there is a hypernymy relation. In kth layer of $\boldsymbol{T}$, we calculate a score s^k as:

$$s^k = \boldsymbol{p}^T \boldsymbol{T}^k \boldsymbol{h} \tag{1}$$

where $\boldsymbol{T}^k$ is matrix and $k \in [1, K]$. Combining K scores, we obtain the similarity vector $\boldsymbol{s}$. After apply a fully connected layer with sigmoid activation function, we get the final probability y:

$$y = \sigma(\boldsymbol{W}\boldsymbol{s} + \boldsymbol{b}) \tag{2}$$

4.2.3 Active learning. Since labeling a large number of hyponym-hypernym pairs for each domain clearly does not scale, we adopt *active learning* as a more guided approach to select examples to label so that we can economically learn an accurate model by reducing the annotation cost. It is based on the premise that a model can get better performance if it is allowed to prepare its own training data, by choosing the most beneficial data points and querying their annotations from annotators. We propose an uncertainty and high confidence sampling strategy (UCS) to select samples which

can improve model effectively. The iterative active learning algorithm is shown in Algorithm 1.

Algorithm 1 UCS active learning algorithm

Input: unlabeled dataset D, test dataset T, scoring function $f(\cdot,\cdot)$, human labeling H, the number of human labeling samples in each iteration K; **Output**: scoring function $\hat{f}(\cdot,\cdot)$, predict score S

1: **procedure** AL(D, D_0, T, f, H, K)
2: $i \leftarrow 0$
3: $D_0 \leftarrow random\_select(D, K)$
4: $L_0 \leftarrow H(D_0)$
5: $D \leftarrow D - D_0$
6: $\hat{f}, fs \leftarrow train\_test(f, L_0, T)$
7: $S \leftarrow \hat{f}(D)$
8: **repeat**
9: $p_i = \frac{|S_i - 0.5|}{0.5}$
10: $D_{i+1} \leftarrow D(Top(p_i, \alpha K)) \bigcup D(Bottom(p_i, (1-\alpha)K))$
11: $L_{i+1} \leftarrow H(D_{i+1}) \bigcup L_i$
12: $D \leftarrow D - D_0$
13: $\hat{f}, fs \leftarrow train\_test(f, L_{i+1}, T)$
14: $S \leftarrow \hat{f}(D)$
15: **until** fs not improves in n step
16: **end procedure**

As line 3 to 7 show, we first randomly select a dataset D_0 which contains K samples from the unlabeled dataset D and ask domain experts to label the samples from D_0. As a result, we obtain the initial labeled dataset L_0 and D_0 is removed from the D. Then, we train the projection learning model f using L_0 and test the performance on the test dataset T. fs is the metrics on T. At last, we predict the unlabeled dataset D using the trained $\hat{f}$ and get the score S_0.

Next, we iteratively select unlabeled samples to label and use them to enhance our model. We propose an active learning sampling strategy named uncertainty and high confidence sampling (UCS) which select unlabeled samples from two factors. The first factor is based on classical uncertainty sampling (US) [**?**]. If the prediction score of a sample is close to 0.5, it means the current model is difficult to judge the label of this sample. If the expert labels this example, the model can enhance its ability by learning this sample. We calculate this probability by $\frac{|S_i - 0.5|}{0.5}$ in line 9. Besides, we believe those samples with high confidence are also helpful in the task of hypernym discovery, since the model is likely to predict some difficult negative samples as positive with high confidence when encountering relations such as *same_as* or *similar*. The signal from human labeling can correct this problem in time. Thus, we select those samples with high scores as well in line 10. In addition, we utilize α to control the weight of different sampling size. Then, we get the new human labeled dataset which can be used to train a better model. As a result, with the number of labeled data increases, the performance of our model will also increase.

Finally, this iterative process will be stopped when the performance of the model fs does not increase in n rounds. During the process, we not only get a better model but also reduce the cost of human labeling.

5 E-COMMERCE CONCEPTS

In the layer of e-commerce concepts, each node represents a specific shopping scenario, which can be interpreted by at least one primitive concept. In this section, we first introduce the high criteria of a good e-commerce concept using several examples, then show how we generate all those e-commerce concepts and further propose an algorithm to link e-commerce concepts to the layer of primitive concepts.

5.1 Criteria

As introduced in Section 2, user needs are conceptualized as e-commerce concepts in AliCoCo, and a good e-commerce concept should satisfy the following criteria:

(1) E-commerce Meaning. It should let anyone easily think of some items in the e-commerce platform, which means it should naturally represent a particular shopping need. Phrases like "blue sky" or "hens lay eggs" are not e-commerce concepts, because we can hardly think of any related items.

(2) Coherence. It should be a coherent phrase. Counter-examples can be "gift grandpa for Christmas" or "for kids keep warm", while the coherent ones should be "Christmas gifts for grandpa" and "keep warm for kids".

(3) Plausibility. It should be a plausible phrase according to commonsense knowledge. Counter-examples can be "sexy baby dress" or "European Korean curtain" since we humans will not describe a dress for babies using the word "sexy" and a curtain can not be in both European style and Korean style.

(4) Clarity. The meaning of an e-commerce concept should be clear and easy to understand. Counter-examples can be "supplementary food for children and infants" where the subject of this can be either older-aged children or newborns. This may lead to a confusion for our customers.

(5) Correctness. It should have zero pronunciation or grammar error.

5.2 Generation

There is no previous work on defining such e-commerce concepts and few on mining such phrases from texts. In practice, we propose a two-stage framework: firstly we use two different ways to generate large amount of possible e-commerce concept candidates, then a binary classification model is proposed to identify those concepts which satisfy our criteria.

5.2.1 Candidate Generation. There are two different ways to generate concept candidates. The first is mining raw concepts

from texts. In practice, we adopt AutoPhrase[25] to mine possible concept phrases from large corpora in e-commerce including search queries, product titles, user-written reviews and shopping guidance written by merchants. Another alternative is to generating new candidates using existing primitive concepts[4]. For example, we combine "*Location: Indoor*" with "*Event: Barbecue*" to get a new concept "indoor barbecue", which is not easy to be mined from texts since it's a little bit unusual. However, it is actually a quite good e-commerce concept since one goal of AliCoCo is to cover as many user needs as possible. The rule to combine different classes of primitive concepts is using some automatically mined then manually crafted patterns. For example, we can generate a possible concept "warm hat for traveling" using a pattern "[*class: Function*] [*class: Category*] for [*class: Event*]". Table 1 shows some patterns used in practice and corresponding e-commerce concepts, including some bad ones waiting to be filtered out in the following step.

5.2.2 Classification. To automatically judge whether a candidate concept satisfies the criteria of being a qualified e-commerce concept or not, the main challenge is to test its plausibility. For the other four criteria, character-level and word-level language models and some heuristic rules are able to meet the goal. However, it is difficult for machines to grasp commonsense knowledge as we humans do to know that "sexy" is not suitable to describe a dress when it's made for a child. Moreover, the lack of surrounding contexts makes the problem more challenging, since our concepts are too short (2-3 words on average).

To tackle this problem, we propose a knowledge-enhanced deep classification model to first link each word of a concept to an external knowledge base then introduce rich semantic information from it. The model architecture is shown in Figure 5, which is based on Wide&Deep [7] framework. The input is a candidate concept c, and the output is a score, measuring the probability of c being a good e-commerce concept. In this paper, we denote a char as a single Chinese or English character, and a segmented word (or term) is a sequence of several chars such as "Nike" or "牛仔裤 (jeans)". We perform Chinese word segmentation for all the input concepts before feeding to the model.

In the Deep side, there are mainly two components. Firstly, a char level BiLSTM is used to encode the candidate concept c by feeding the char-level embedding sequence $\{\boldsymbol{ch}_1, \boldsymbol{ch}_2, ...\boldsymbol{ch}_n\}$ after simple embedding lookup. After mean pooling, we get the concept embedding $\boldsymbol{c}_1$. The other component is knowledge-enhanced module. The input consists of there parts: 1) pre-trained word embeddings; 2) POS tag [28] embedding using a lookup table; 3) NER label [11] embedding using a

[4]If a primitive concept satisfies all five criteria, it can be regarded as an e-commerce concept as well.

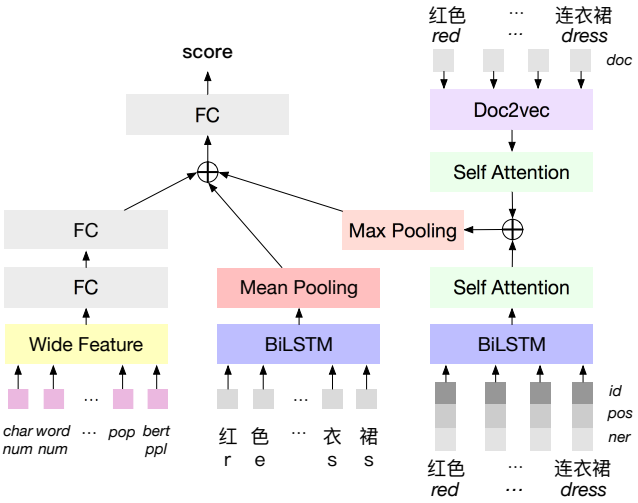

Figure 5: Overview of knowledge-enhanced deep model for e-commerce concept classification.

lookup table. After concatenate those three embeddings, we obtain the input embedding sequence of candidate concept c: $\{\boldsymbol{w}_1, \boldsymbol{w}_2, ...\boldsymbol{w}_m\}$ $(m < n)$. After going through BiLSTM, we use self attention mechanism [29] to further encode the mutual influence of each word within the concept and get a sequence output $\{\boldsymbol{w'}_1, \boldsymbol{w'}_2, ...\boldsymbol{w'}_m\}$. To introduce external knowledge into the model to do commonsense reasoning on short concepts, we link each word w to its corresponding Wikipedia article if possible. For example, "性感 (sexy)" can be linked to https://zh.wikipedia.org/wiki/%E6%80%A7%E6%84%9F (https://en.wikipedia.org/wiki/Sexy). Then we extract the gloss of each linked Wikipedia articl as the external knowledge to enhance the feature representation of concept words. A gloss is a short document to briefly introduce a word. We employ Doc2vec [16] to encode each extracted gloss for word $\boldsymbol{w}_i$ as $\boldsymbol{k}_i$. Then, we get the representation of the knwoledge sequence after a self attention layer: $\{\boldsymbol{k'}_1, \boldsymbol{k'}_2, ...\boldsymbol{k'}_m\}$. We concatenate $\boldsymbol{w'}_i$ as $\boldsymbol{k'}_i$ and use max-pooling to get the final knowledge-enhanced representation of candidate concept $\boldsymbol{c}_2$.

In the Wide side, we mainly adopt pre-calculated features such as the number of characters and words of candidate concept, the perplexity of candidate concept calculated by a BERT [10] model specially trained on e-commerce corpus, and other features like the popularity of each word appearing in e-commerce scenario. After going through two fully connected layers, we get the wide feature representation $\boldsymbol{c}_3$.

The final score $\hat{y}_c$ is calucalated by concatenating the three embedding $\boldsymbol{c}_1$, $\boldsymbol{c}_2$ and $\boldsymbol{c}_3$ then going through a MLP layer. We use point-wise learning with the negative log-likelihood

Pattern	*Good Concept*	*Bad Concept*
[*class: Function*] [*class: Category*] for [*class: Event*]	warm hat for traveling	warm shoes for swimming
[*class: Style*] [*class: Time->Season*] [*class: Category*]	British-style winter trench coat	casual summer coat
[*class: Location*] [*class: Event->Action*] [*class: Category*]	British imported snacks	Bird's nest imported from Ghan
[*class: Function*] for [*class: Audience->Human*]	health care for olds	waterproofing for middle school students
[*class: Event->Action*] in [*class: Location*]	traveling in European	Bathing in the classroom

Table 1: Some patterns used to generate e-commerce concepts.

objective function to learn the parameters of our model:

$$\mathscr{L} = -\sum_{(c)\in D^+} \log \hat{y}_c + \sum_{(c)\in D^-} \log(1-\hat{y}_c) \tag{3}$$

where D^+ and D^- are the good and bad e-commerce concepts.

We expect this model can help filter out most of bad candidate concepts generated in the first step. To strictly control the quality, we randomly sample a small portion of every output batch which passes the model checking to ask domain experts to manually annotate. Only if the accuracy riches a certain threshold, the whole batch of concepts will be added into AliCoCo. Besides, the annotated samples will be added to training data to iteratively improve the model performance.

5.3 Understanding

For those good e-commerce concepts which are directly mined from text corpora, they are isolated phrases waiting to be integrated into AliCoCo. To better understand (or interpret) those user needs (aka. e-commerce concepts), it is a vital step to link them to the layer of primitive concepts. We call the main task as *"e-commerce concept tagging"*. Revisit the example shown in Section 2, given an surface from "outdoor barbecue", we need to infer that "outdoor" is a "*Location*" and "barbecue" is an "*Event*". However, word "barbecue" can also be a movie in the layer of primitive concepts, so it may be recognized into the class of "*IP*". We formulate this task as a *short text* Named Entity Recognition (NER) problem, which is more challenging to a normal NER task since concept phrases here are too short (2-3 words on average). Lack of contextual information make it harder to disambiguate between different classes.

To overcome the above challenges, we propose a text-augmented deep NER model with fuzzy CRF, shown in Figure 6. The input of this task is a sequence of concept word $\{\boldsymbol{w}_1, \boldsymbol{w}_2, ...\boldsymbol{w}_m\}$ after Chinese word segmentation, while the output is a sequence of same length $\{\boldsymbol{y}_1, \boldsymbol{y}_2, ...\boldsymbol{y}_m\}$ denoting the class labels for each word with In/Out/Begin (**I/O/B**) scheme. The model consisting of two components: text-augmented concept encoder and fuzzy CRF layer.

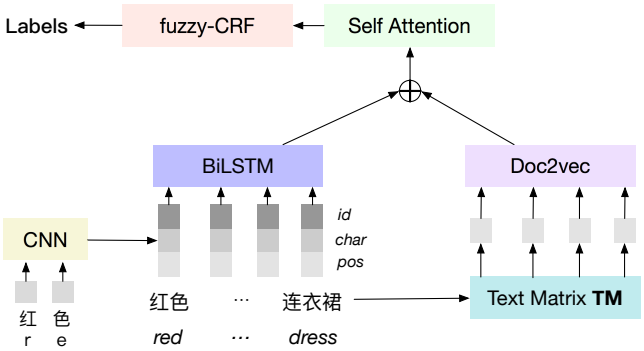

Figure 6: Overview of text-augmented deep NER model for e-commerce concept tagging.

5.3.1 Text-augmented concept encoder. To leverage informative features in the representation layer, we employ word-level, char-level features and position features. We randomly initialize a lookup table to obtain an embedding for every character. Let C be the vocabulary of characters, a word w_i can be represented as a sequence of character vectors: $\{\boldsymbol{c}_1^i, \boldsymbol{c}_2^i, ..., \boldsymbol{c}_t^i\}$, where $\boldsymbol{c}_j^i$ is the vector for the j-th character in the word w_i and t is the word length. Here we adopt a convolutional neural network (CNN) architecture to extract the char-level features $\boldsymbol{c}_i$ for each word w_i. Specifically, we use a convolutional layer with window size k to involve the information of neighboring characters for each character. A max pooling operation is then applied to output the final character representation as follows:

$$\boldsymbol{c}_j^i = \text{CNN}([\boldsymbol{c}_{j-k/2}^i, ..., \boldsymbol{c}_j^i, ..., \boldsymbol{c}_{j+k/2}^i]) \tag{4}$$

$$\boldsymbol{c}_i = \text{MaxPooling}([\boldsymbol{c}_0^i, ...\boldsymbol{c}_j^i, ...]) \tag{5}$$

To capture word-level features, we use pre-trained word embeddings from GloVe [22] to map a word into a real-valued vector $\boldsymbol{x}_i$, as the initialized word features and will be fine-tuned during training. Furthermore, we calculate part-of-speech tagging features $\boldsymbol{p}_i$. Finally, we obtain the word representation $\boldsymbol{w}_i$ by concatenating three embeddings:

$$\boldsymbol{w}_i = [\boldsymbol{x}_i; \boldsymbol{c}_i; \boldsymbol{p}_i]. \tag{6}$$

Similar to the classification model introduced in the previous task, we feed the sequence of word representations to BiLSTM layer to obtain hidden embeddings $\{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_m\}$. To augment our model with more textual information, we

construct a textual embedding matrix **TM** by mapping each word back to large-scale text corpus to extract surrounding contexts and encode them via Doc2vec. Thus, we lookup each word w_i in **TM** to obtain a text-augmented embedding $\boldsymbol{tm}_i$. We concatenate $\boldsymbol{h}_i$ and $\boldsymbol{tm}_i$ then use a self attention layer to adjust the representations of each words by considering the augmented textual embeddings of surrounding words, aiming to obtain better feature representations for this task:

$$\boldsymbol{h'}_i = \text{SelfAtt}([\boldsymbol{h}_i; \boldsymbol{tm}_i]). \tag{7}$$

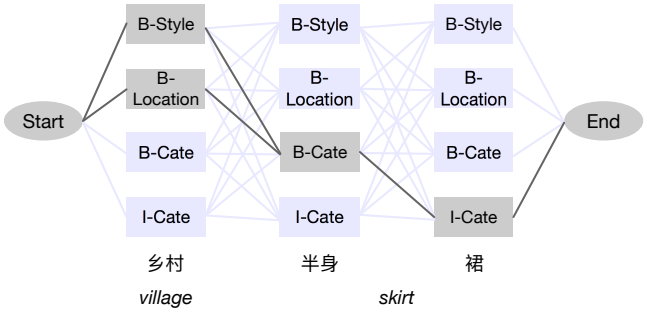

Figure 7: A real example in fuzzy CRF layer.

5.3.2 Fuzzy CRF layer. Following the concept encoding module, we feed the embeddings to a CRF layer. Different from normal CRF, we use a fuzzy CRF [26] to better handle the disambiguation problem since the valid class label of each word is not unique and this phenomenon is more severe in this task since our concept is too short. Figure 7 shows an example, where the word "乡村 (village)" in the e-commerce concept "乡村半身裙 (village skirt)" can linked to the primitive concept "空间: 乡村 *(Location: Village)*" or "风格: 乡村 *(Style: Village)*". They both make sense. Therefore, we adjust the final probability as

$$L(y|\boldsymbol{X}) = \frac{\sum_{\hat{y} \in Y_{possible}} e^{s(X,\hat{y})}}{\sum_{\hat{y} \in Y_X} e^{s(X,\hat{y})}}. \tag{8}$$

where Y_X means all the possible label sequences for sequence X, and $Y_{possible}$ contains all the possible label sequences.

6 ITEM ASSOCIATION

Items are the most essential nodes in any e-commerce knowledge graph, since the ultimate goal of e-commerce platform is to make sure that customers can easily find items that satisfy their needs. So far, we conceptualize user needs as e-commerce concepts and interpret them using the structured primitive concepts. The last thing is to associate billions of items in Alibaba with all the concepts (both primitive and e-commerce) to form the complete AliCoCo.

Since primitive concepts are similar to single-value tags and properties, the mapping between primitive concepts and items are relatively straightforward. Therefore, in this section, we mainly introduce the methodology of associating items with e-commerce concepts, where the latter ones representing certain shopping scenarios usually carry much more complicated semantic meanings. Besides, the association between an e-commerce concept and certain items can not be directly inferred from the association between corresponding primitive concepts and their related items due to a phenomenon called "semantic drift". For example, charcoals are necessary when we want to hold an "outdoor barbecue", however, they have nothing to do with primitive concept "*Location: Outdoor*".

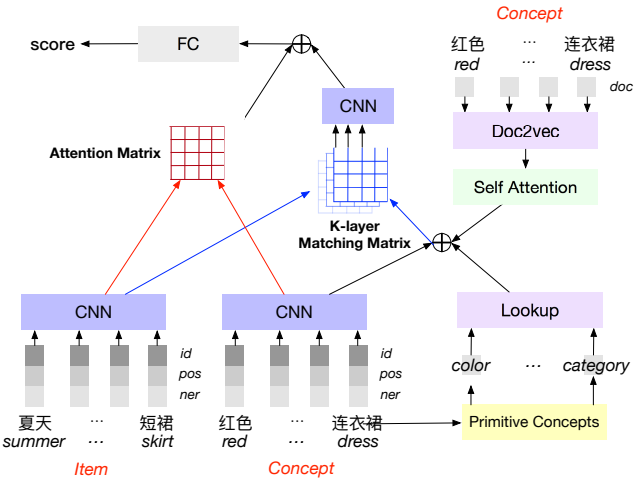

Figure 8: Overview of knowledge-aware deep semantic matching model for association between e-commerce concepts and items.

We formulate this task as *semantic matching* between texts [13, 21, 31], since we only use textual features of items at current stage. The main challenge to associate e-commerce concepts with related items is that the length of the concept is too short so that limited information can be used. Due to the same reason, there is a high risk that some of less important words may misguide the matching procedure. To tackle it, we propose a knowledge-aware deep semantic matching model shown in Figure 8. The inputs are a sequence of concept words and a sequence of words from the title of a candidate item. We obtain input embeddings concatenating pre-trained word embeddings of two sequences with their POS tag embedding and NER tag embedding (similar to Section 5.3): $\{\boldsymbol{w}_1, \boldsymbol{w}_2, ...\boldsymbol{w}_m\}$ and $\{\boldsymbol{t}_1, \boldsymbol{t}_2, ...\boldsymbol{t}_l\}$. we adopt wide CNNs with window size k to encode the concept and item respectively:

$$\boldsymbol{w'}_i = \text{CNN}([\boldsymbol{w}_{i-k/2}, ..., \boldsymbol{w}_i, ..., \boldsymbol{w}_{i+k/2}]) \tag{9}$$

$$\boldsymbol{t'}_i = \text{CNN}([\boldsymbol{t}_{i-k/2}, ..., \boldsymbol{t}_i, ..., \boldsymbol{t}_{i+k/2}]) \tag{10}$$

Intuitively, different words in the concept should share different weights when matching to the item, and vice versa. Therefore, we apply attention mechanism [3, 19] in our model. An attention matrix is used to model the two-way interactions simultaneously. The values of attention matrix are defined as below:

$$att_{i,j} = \boldsymbol{v}^T \tanh(\boldsymbol{W}_1 \boldsymbol{w'}_i + \boldsymbol{W}_2 \boldsymbol{t'}_j) \tag{11}$$

where $i \in [1, m]$ and $j \in [1, l]$, $\boldsymbol{v}$, $\boldsymbol{W}_1$ and $\boldsymbol{W}_1$ are parameters. Then the weight of each concept word w_i and title word t_i can be calculated as:

$$\alpha_{wi} = \frac{exp(\sum_j att_{i,j})}{\sum_i exp(\sum_j att_{i,j})} \tag{12}$$

$$\alpha_{tj} = \frac{exp(\sum_i att_{i,j})}{\sum_j exp(\sum_i att_{i,j})} \tag{13}$$

Then, we obtain concept embedding $\boldsymbol{c}$ as:

$$\boldsymbol{c} = \sum_i \alpha_{wi} \boldsymbol{w'}_i \tag{14}$$

and item embedding $\boldsymbol{i}$ similarly.

To introduce more informative knowledge to help semantic matching, we obtain the same knowledge embedding sequence in Section 5.2.2:

$$\boldsymbol{k}_i = \text{Doc2vec}(Gloss(w_i)) \tag{15}$$

Besides, we obtain class label id embedding $\boldsymbol{cls}_j$ of jth primitive concept linked with current e-commerce concept. Thus, there are three sequences on the side of concept:

$$\{\boldsymbol{kw}_i\} = \{\boldsymbol{kw}_1, \boldsymbol{kw}_2, ...\boldsymbol{kw}_{2*m+m'}\} =$$
$$\{\boldsymbol{w}_1, \boldsymbol{w}_2, ...\boldsymbol{w}_m, \boldsymbol{k}_1, \boldsymbol{k}_2, ...\boldsymbol{k}_m, \boldsymbol{cls}_1, \boldsymbol{cls}_2, ...\boldsymbol{cls}_{m'}\}$$

where m' is the number of primitive concepts. In the side of item, we directly use the sequence of word embedding $\{\boldsymbol{t}_i\} = \{\boldsymbol{t}_1, \boldsymbol{t}_2, ...\boldsymbol{t}_l\}$. Then, we adopt the idea of Matching Pyramid [21], the values of matching matrix in kth layer are defined as below:

$$match^k_{i,j} = \boldsymbol{kw}_i^T \boldsymbol{W}_k \boldsymbol{t}_j \tag{16}$$

where $i \in [1, 2*m+m']$ and $j \in [1, l]$. Each layer of matching matrix are then fed to 2-layer of CNNs and max-pooling operation to get a matching embedding $\boldsymbol{ci}^k$. The final embedding of matching pyramid $\boldsymbol{ci}$ is obtained by:

$$\boldsymbol{ci} = \text{MLP}([;\boldsymbol{ci}^k;]) \tag{17}$$

The final score measuring the probability is calculated as:

$$score = \text{MLP}([\boldsymbol{c}; \boldsymbol{i}; \boldsymbol{ci}]) \tag{18}$$

7 EVALUATIONS

In this section, we first give a statistical overview of AliCoCo. Next we present experimental evaluations for five main technical modules during the construction of AliCoCo.

7.1 Overall Evaluation

Table 2 shows the statistics of AliCoCo. There are 2,853,276 primitive concepts and 5,262,063 e-commerce concepts in total at the time of writing. There are hundreds of billions of relations in AliCoCo, including 131,968 isA relations within *Category* in the layer of primitive concepts and 22,287,167 isA relations in the layer of e-commerce concepts. For over 3 billion items in Alibaba, 98% of them are linked to AliCoCo. Each item is associated with 14 primitive concepts and 135 e-commerce concepts on average. Each e-commerce concept is associated with 74,420 items on average. The number of relations between e-commerce concept layer and primitive concept layer is 33,495,112.

AliCoCo is constructed semi-automatically. For those nodes and relations mined by models, we will randomly sample part of data and ask human annotators to label. Only if the accuracy achieves certain threshold, the mined data will be added into AliCoCo to ensure the quality. Besides, for those dynamic edges (associated with items), we monitor the data quality regularly.

Overall			
# Primitive concepts		2,853,276	
# E-commerce concepts		5,262,063	
# Items		> 3 billion	
# Relations		> 400 billion	
Primitive concepts			
# Audience	# Brand	# Color	# Design
15,168	879,311	4,396	744
# Event	# Function	# Category	# IP
18,400	16,379	142,755	1,491,853
# Material	# Modifier	# Nature	# Organization
4,895	106	75	5,766
# Pattern	# Location	# Quantity	# Shape
486	267,359	1,473	110
# Smell	# Style	# Taste	# Time
9,884	1,023	138	365
Relations			
# IsA in primitive concepts		131,968 (only in *Category*)	
# IsA in e-commerce concepts		22,287,167	
# Item - Primitive concepts		21 billion	
# Item - E-commerce concepts		405 billion	
# E-commerce - Primitive cpts		33,495,112	

Table 2: Statistics of AliCoCo at the time of writing.

To evaluate the coverage of actual shopping needs of our customers, we sample 2000 search queries at random and manually rewrite them into coherent word sequences, then we search in AliCoCo to calculate the coverage of those words. We repeat this procedure every day, in order to detect new trends of user needs in time. AliCoCo covers over 75% of shopping needs on average in continuous 30 days, while

this number is only 30% for the former ontology mentioned in Section 1.

7.2 Primitive Concept Mining

After defining 20 different domains in the taxonomy, we quickly enlarge the size of primitive concepts by introducing knowledges from several existing structured or semi-structured knowledge bases in general-purpose domain. During this step, vocabulary sizes of domains such as *Location*, *Organization* and *IntellectulProperty* can be quickly enlarged. Other domains are for e-commerce use, and we mainly leverage the existing e-commerce semi-structured data: CPV, since most of *Property*s can be matched to our domains such as *Brand*, *Color*, *Material*, etc.

After rule based alignments and cleaning, around $2M$ primitive concepts can be drawn from multiple sources. We adopt the idea of distant supervision to generate a large amount of training samples, in order to mine new concepts. We use a dynamic programming algorithm of max-matching to match words in the text corpora and then assign each word with its domain label in IOB scheme using existing primitive concepts. We filter out sentences whose matching result is ambiguous and only reserve those that can be perfectly matched (all words can be tagged by only one unique label) as our training data. We generate around $6M$ training data in this way. In each epoch of processing $5M$ sentences, our mining model is able to discover around $64K$ new candidate concepts on average. After manually checking the correctness by crowdsourcing services, around $10K$ correct concepts can be added into our vocabulary in each round. The mining procedure is continuously running, and the total number of primitive concepts from all 20 domains is 2,758,464 at the time of writing.

7.3 Hypernym Discovery

In order to organize all the primitive concepts into a fine-grained taxonomy, we propose an active learning framework to iteratively discover isA relation between different primitive concepts. To demonstrate the superior of our framework, we perform several experiments on a ground truth dataset collected after the taxonomy is constructed. We randomly sample 3,000 primitive concepts in the class of "*Category*" which have at least one hypernym, and retrieve 7,060 hyponym-hypernym pairs as positive samples. We split the positive samples into training / validation / testing sets (7:2:1). The search space of hypernym discovery is actually the whole vocabulary, making the number and quality of negative samples very important in this task. The negative samples of training and validation sets are automatically generated from positive pairs by replacing the hypernym of each pair with a random primitive concept from "*Category*" class. In the following experiments, mean average precision (MAP), mean reciprocal rank (MRR) and precision at rank 1 (P@1) are used as evaluation metrics.

To verify the appropriate number of negative samples for each hyponym during training, we perform an experiment shown in Figure 9(left), where N in x-axis represents the ratio of negative samples over positive samples for each hyponym. The results indicate different size of negative samples influence the performance differently. As N gradually increases, the performance improves and achieves best around 100. Thus, we construct the candidate training pool in the following active learning experiment with a size of 500,000.

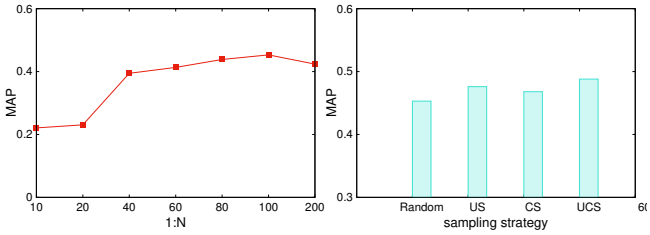

Figure 9: Left: the influence of different negative sample sizes in hypernym discovery on test set. Right: the best performance of different sampling strategies in active learning.

Table 3 shows experimental results of different sampling strategies during our active learning framework, where *Random* means training using the whole candidate pool without active learning. We set the select data size K as 25,000 in each iteration as mentioned in Section 4.2. When it achieves similar MAP score in four active learning strategies, we can find that all the active learning sampling strategies can reduce labeled data to save considerable manual efforts. UCS is the most economical sampling strategy, which only needs $325k$ samples, reducing 35% samples comparing to random strategy. It indicates that high confident negative samples are also important in the task of hypernym discovery.

Strategy	Labeled Size	MRR	MAP	P@1	Reduce
Random	500k	58.97	45.30	45.50	-
US	375k	59.66	45.73	46.00	150k
CS	400k	58.96	45.22	45.30	100k
UCS	325k	59.87	46.32	46.00	175k

Table 3: Experimental results of different sampling strategy in hypernym discovery.

In Figure 9 (right), we show the best performance of each sampling strategies during the whole training procedure.

UCS outperforms other three strategies and achieves a highest MAP of 48.82%, showing the importance of selecting the most valuable samples during model training.

7.4 E-commerce Concept Classification

In this subsection, we mainly investigate how each component of our model influences the performance in the task of judging whether a candidate e-commerce concept satisfy the criteria or not (Section 5.2.2).

We randomly sample a large portion of e-commerce concepts from the candidate set and ask human annotators to label [5]. The final dataset consists of $70k$ samples (positive: negative= 1: 1). Then we split the dataset into 7:1:2 for training, validation and testing.

Model	Precision
Baseline (LSTM + Self Attention)	0.870
+Wide	0.900
+Wide & BERT	0.915
+Wide & BERT & Knowledge	**0.935**

Table 4: Experimental results in shopping concept generation.

Results of ablation tests are shown in Table 4. Comparing to the baseline, which is a base BiLSTM with self attention architecture, adding wide features such as different syntactic features of concept improves the precision by 3% in absolute value. When we replace the input embedding with BERT output, the performance improves another 1.5%, which shows the advantage of rich semantic information encoded by BERT. After introducing external knowledge into our model, the final performance reaches to 0.935, improving by a relative gain of 7.5% against the baseline model, indicating that leveraging external knowledge benefits commonsense reasoning on short concepts.

7.5 E-commerce Concept Tagging

To associate those e-commerce concepts which are directly mined from text corpus to the layer of primitive concepts, we propose a text-augmented NER model with fuzzy CRF mentioned in Section 5.3 to link an e-commerce concept to its related primitive concepts. We randomly sample a small set $(7,200)$ of e-commerce concepts and ask human annotators to label the correct class labels for each primitive concepts within the e-commerce concepts. To enlarge the training data, we use the similar idea of distant supervision mentioned in Section 7.2 to automatically generate $24,000$ pairs of data. Each pair contains a compound concept and its corresponding gold sequence of domain labels. We split $7,200$ pairs of manually labeled data into $4,800/1,400/1,000$ for training, validation and testing. $24,000$ pairs of distant supervised data are added into training set to help learn a more robust model.

Model	Precision	Recall	F1
Baseline	0.8573	0.8474	0.8523
+Fuzzy CRF	0.8731	0.8665	0.8703
+Fuzzy CRF & Knowledge	**0.8796**	**0.8748**	**0.8772**

Table 5: Experimental results in shopping concept tagging.

Experimental results are shown in Table 5. Comparing to baseline which is a basic sequence labeling model with Bi-LSTM and CRF, adding *fuzzy CRF* improves 1.8% on F1, which indicates such multi-path optimization in CRF layer actually contributes to disambiguation. Equipped with external knowledge embeddings to further enhance the textual information, our model continuously outperform to 0.8772 on F1. It demonstrates that introducing external knowledge can benefit tasks dealing with short texts with limited contextual information.

7.6 Concept-Item Semantic Matching

In this subsection, we demonstrate the superior of our semantic matching model for the task of associating e-commerce concepts with billion of items in Alibaba. We create a dataset with a size of $450m$ samples, among which $250m$ are positive pairs and $200m$ are negative pairs. The positive pairs comes from strong matching rules and user click logs of the running application on Taobao mentioned in Section 1. Negative pairs mainly comes from random sampling. For testing, we randomly sample 400 e-commerce concepts, and ask human annotator to label based on a set of candidate pairs. In total, we collect $200k$ positive pairs and $200k$ negative pairs as testing set.

Model	AUC	F1	P@10
BM25	-	-	0.7681
DSSM [13]	0.7885	0.6937	0.7971
MatchPyramid [21]	0.8127	0.7352	0.7813
RE2 [31]	0.8664	0.7052	0.8977
Ours	0.8610	0.7532	0.9015
Ours + Knowledge	**0.8713**	**0.7769**	**0.9048**

Table 6: Experimental results in semantic matching between e-commerce concepts and items.

Table 6 shows the experimental result, where F1 is calculated by setting a threshold of 0.5. Our knowledge-aware

[5]The annotation task lasts for several months until we get enough training samples.

deep semantic matching model outperforms all the baselines in terms of AUC, F1 and Precision at 10, showing the benefits brought by external knowledge. To further investigate how knowledge helps, we dig into cases. Using our base model without knowledge injected, the matching score of concept "中秋节礼物 (Gifts for Mid-Autumn Festival)" and item "老式大月饼共800g云南特产荞三香大荞饼荞酥散装多口味 (Old big moon cakes 800g Yunnan...)" is not confident enough to associate those two, since the texts of two sides are not similar. After we introduce external knowledge for "中秋节 (Mid-Autumn Festival)" such as "中秋节自古便有赏月、吃月饼、赏桂花、饮桂花酒等习俗。(It is a tradition for people to eat moon cakes in Mid-Autumn...)", the attention score for "中秋节 (Mid-Autumn Festival)" and "月饼 (moon cakes)" increase to bridge the gap of this concept-item pair.

8 APPLICATIONS

AliCoCo has already supported a series of downstream applications in Alibaba's ecosystem, especially in search and recommendation, two killer applications in e-commerce. In this section, we introduce some cases we already succeed, those we are attempting now, and some other we would like to try in the future.

8.1 E-commerce Search

8.1.1 Search relevance. Relevance is the core problem of a search engine, and one of the main challenges is the vocabulary gap between user queries and documents. This problem is more severe in e-commerce since language in item titles is more professional. Semantic matching is a key technique to bridge the gap in between to improve relevance. IsA relations is important in semantic matching. For example, if a user search for a "top", search engine may classify those items whose title only contains "jacket" but without "top" as irrelevance. Once we have the prior knowledge that "jacket is a kind of top", this case can be successfully solved. Comparing to a former category taxonomy, which only has 15k different category words and 10k isA relations, AliCoCo containing 10 times categories words and isA relations. Offline experiments show that our data improves the performance of the semantic matching model by 1% on AUC, and online tests show that the number of relevance bad cases is dropped by 4%, meaning user satisfaction is improved.

8.1.2 Semantic search & question answering. As shown in Figure 2(a), semantic search empowered by AliCoCo is ongoing at the time of writing. Similar to searching "China" on Google and then getting a knowledge card on the page with almost every important information of China, we are now designing a more structured way to display the knowledge of "Tools you need for baking" once a customer searching for "baking". On the other hand, this application requires a high accuracy and recall of relations, which are still sparse in the current stage of AliCoCo. Question answering is a way of demonstrating real intelligence of a search engine. Customers are used to keyword based search for years in e-commerce. However, at some point we may want to ask an e-commerce search engine "What should I prepare for hosting next week's barbecue?". We believe AliCoCo is able to provide ample imagination towards this goal with continuous efforts to integrate more knowledge especially concerning common sense.

8.2 E-commerce Recommendation

8.2.1 Cognitive recommendation. As we introduce in Section 1, a natural application of e-commerce concepts is directly recommending them to users together with its associated items. In the snapshot shown in Figure 2(b), concept "Tools for Baking" is displayed as a card, with the picture of a representative item. Once users click on this card, it jumps to a page full of related items such as egg scrambler and strainer. We perform thorough offline and online experiments in a previous work [18]. It has already gone into production for more than 1 year with high click-through rate and satisfied GMV (Gross Merchandise Value). According to a survey conducted by online users, this new form of recommendation brings more novelty and further improve user satisfaction. This application is totally based on the complete functionality of AliCoCo, which demonstrates its great value and potential.

8.2.2 Recommendation reason. The advantages of e-commerce concepts include its clarity and brevity, which make them perfect recommendation reasons to display when recommending items to customers. This idea is currently experimented at the time of writing.

9 RELATED WORK

Great human efforts have been devoted to construct open domain KGs such as Freebase [5] and DBpedia [2], which typically describe specific facts with well-defined type systems rather than inconsistent concepts from natural language texts. Probase [30] constructs a large-scale probabilistic taxonomy of concepts, organizing general concepts using isA relations. Different from AliCoCo, concepts in Probase do not have classes so that semantic heterogeneity is handled implicitly. From this perspective, the structure of AliCoCo is actually more similar to KGs with a type system such as Freebase. ConceptNet [27] tries to include common sense knowledge by recognizing informal relations between concepts, where the concepts could be the conceptualization of any human knowledge such as "games with a purpose"

appearing in free texts. Inspired by the construction of open-domain KGs, different kinds of KGs in e-commerce are constructed to describe relations among users, items and item attributes [1, 6]. One famous example is the "Product Knowledge Graph" (PG) [6] of Amazon, another e-commerce giant in the world. The major difference is that they do not focus on user needs as we do. In AliCoCo, we formally define user needs and introduce a new type of nodes named e-commerce concepts to explicitly represent various shopping needs and further link them to the layer of primitive concepts for semantic understanding. Although we do not discuss much, AliCoCo can be connected to open-domain KGs through the layer of primitive concepts (e.g. IP, Organization, etc) just like PG, making it more powerful.

10 CONCLUSION

In this paper, we point out that there is a huge semantic gap between user needs and current ontologies in most e-commerce platforms. This gap inevitably leads to a situation where e-commerce search engine and recommender system can not understand user needs well, which, however, are precisely the ultimate goal of e-commerce platforms try to satisfy. To tackle it, we introduce a specially designed e-commerce cognitive concept net "AliCoCo" practiced in Alibaba, trying to conceptualize user needs as various shopping scenarios, also known as "e-commerce concepts". We present the detailed structure of AliCoCo and introduce how it is constructed with abundant evaluations. AliCoCo has already benefited a series of downstream e-commerce applications in Alibaba. Towards a subsequent version, our future work includes: 1) Complete AliCoCo by mining more unseen relations containing commonsense knowledge, for example, "boy's T-shirts" implies the "*Time*" should be "*Summer*", even though term "summer" does not appear in the concept. 2) Bring probabilities to relations between concepts and items. 3) Benefit more applications in e-commerce or even beyond e-commerce.

11 ACKNOWLEDGMENT

We deeply thank Mengtao Xu, Yujing Yuan, Xiaoze Liu, Jun Tan and Muhua Zhu for their efforts on the construction of AliCoCo.

REFERENCES

[1] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation. *arXiv preprint arXiv:1805.03352* (2018).

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. *Dbpedia: A nucleus for a web of open data.* Springer.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[4] Paul Bloom. 2003. Glue for the mental world. *Nature* 421, 6920 (2003), 212.

[5] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*. 1247–1250.

[6] Rose Catherine, Kathryn Mazaitis, Maxine Eskenazi, and William Cohen. 2017. Explainable entity-based recommendations with knowledge graphs. *arXiv preprint arXiv:1707.05254* (2017).

[7] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10.

[8] Sergio Cleger-Tamayo, Juan M Fernandez-Luna, and Juan F Huete. 2012. Explaining neighborhood-based recommendations. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1063–1064.

[9] Felipe Costa, Sixun Ouyang, Peter Dolog, and Aonghus Lawlor. 2018. Automatic Generation of Natural Language Explanations. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion*. ACM, 57.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[11] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 363–370.

[12] Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, 539–545.

[13] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2333–2338.

[14] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* (2015).

[16] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. 1188–1196.

[] David D Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*. Elsevier, 148–156.

[17] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 345–354.

[18] Xusheng Luo, Yonghua Yang, Kenny Qili Zhu, Yu Gong, and Keping Yang. 2019. Conceptualize and Infer User Needs in E-commerce. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, 2517–2525.

[19] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[6] http://conferences.cis.umac.mo/icde2019/wp-content/uploads/2019/06/icde-2019-keynote-luna-dong.pdf

[20] G Murphy. 2002. TheBig Bookof Concepts. *Cambridge: TheMITPress* (2002).

[21] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[22] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[23] Nils Reimers and Iryna Gurevych. 2017. Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks. *arXiv* (2017).

[24] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. ACM, 285–295.

[25] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1825–1837.

[26] Jingbo Shang, Liyuan Liu, Xiang Ren, Xiaotao Gu, Teng Ren, and Jiawei Han. 2018. Learning named entity tagger using domain-specific dictionary. *arXiv preprint arXiv:1809.03599* (2018).

[27] Robert Speer and Catherine Havasi. 2012. Representing General Relational Knowledge in ConceptNet 5.. In *LREC*. 3679–3686.

[28] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for computational Linguistics, 173–180.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[30] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 481–492.

[31] Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019. Simple and Effective Text Matching with Richer Alignment Features. *arXiv preprint arXiv:1908.00300* (2019).

[32] Markus Zanker and Daniel Ninaus. 2010. Knowledgeable explanations for recommender systems. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, Vol. 1. IEEE, 657–660.

[33] Yongfeng Zhang and Xu Chen. 2018. Explainable Recommendation: A Survey and New Perspectives. *arXiv preprint arXiv:1804.11192* (2018).

[34] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1059–1068.